

## Listing of Claims

1-3. (Cancelled)

4. (Currently Amended) The A method of ~~claim 1~~ reducing the number of times a tree data structure is rebalanced comprising the steps of:

(I) allowing a sub-tree of the tree data structure to grow until a number of unbalanced levels reaches a threshold greater than one; and

(II) rebalancing the tree data structure when the threshold is reached by: ~~wherein the step of rebalancing the tree data structure further comprises:~~

(a) developing first and second sets of rebalancing operation tasks, the first set of operation tasks operable to effect a first set of element state transitions and the second set of operation tasks operable to effect a second set of element state transitions, the first and second set of element state transition being distinct one from the other;

(b) performing the first set of operation tasks in a first phase; and

(c) performing the second set of operation tasks in a second phase.

5-9. (Cancelled)

10. (Original) A method of performing a rebalancing operation upon a tree data structure comprising the steps of:

(a) allowing a sub-tree of the tree data structure to grow unbalanced to a threshold level greater than one;

- 6 (b) developing, in the case where the sub-tree reaches the threshold level, first and  
7 second sets of rebalancing operation tasks, the first and second set of rebalancing  
8 operation tasks operable to effect a first and second set of element state transitions  
9 respectively;
- 10 (c) performing the first set of operation tasks in a first phase; and
- 11 (d) performing the second set of operation tasks in a second phase.

1 11-14. (Cancelled)

1 15. (Previously Presented) A system comprising:

- 2 means for storing a tree data structure;
- 3 means for tracking the execution of operations upon the tree data structure; and
- 4 means for rebalancing the tree data structure when an unbalanced sub-tree of the tree data  
5 structure reaches a threshold level greater than one, the rebalancing including a  
6 first rebalancing phase in which rebalancing operations are executed in parallel  
7 and nodes of the unbalanced sub-tree are unlocked, and a second rebalancing  
8 phase in which different rebalancing operations are executed.

1 16. (Cancelled)

1 17. (Previously Presented) A method of deferring the rebalancing of a tree data

2 structure comprising the steps of:

3 (a) tracking the performance of operations upon the tree data structure; and

4 (b) rebalancing the tree data structure when an unbalanced sub-tree of the tree

5 data structure reaches a threshold level greater than one, the rebalancing

6 further comprising creating a first set of rebalancing operation tasks, the

7 first set of rebalancing operation tasks being characterized by navigation

8 of the tree data structure using at least an existing link, creating a second

9 set of rebalancing operation tasks, the second set of rebalancing operation

10 tasks being different from the first set of rebalancing operation tasks and

11 being characterized by location of elements within the tree data structure

12 using at least one pointer external to the tree data structure and created by

13 the first set of rebalancing operation tasks, and performing at least one

14 operation task of the first set of rebalancing operation tasks in a first phase

15 and at least one of the second set of rebalancing operation tasks in a

16 second phase.

1 18. (Original) A method of deferring the rebalancing of a tree data structure comprising

2 the steps of:

3 (a) tracking the performance of operations upon the tree data structure; and

4 (b) rebalancing the tree data structure when an unbalanced sub-tree of the tree

5 data structure reaches a threshold level greater than one, the rebalancing

6 further comprising executing simultaneous rebalancing operations on the  
7 tree data structure including performing any first phase operation task of  
8 each of the simultaneous rebalancing operations in a first phase using  
9 parallel processes, developing a set of serial rebalancing operations during  
10 the first phase, and performing any second phase operation task of each of  
11 the simultaneous rebalancing operations in a second phase, the second  
12 phase operation task having at least one of the set of serial rebalancing  
13 operations.

1 19. (Previously Presented) A method of rebalancing a tree data structure, the method  
2 comprising:  
3 allowing a sub-tree of the tree data structure to grow unbalanced until a threshold  
4 level is reached;  
5 developing a first set of rebalancing operation tasks, the first set of operation tasks  
6 operable in parallel on one or more unlocked nodes of the tree data  
7 structure during a first phase of the rebalancing;  
8 developing a second set of rebalancing operation tasks during execution of the  
9 first set of rebalancing operation tasks; and  
10 executing the second set of rebalancing operation tasks during a second phase of  
11 the rebalancing.

1 20. (Previously Presented) The method of claim 19, wherein execution of the second set  
2 of rebalancing operation tasks is performed without navigating between nodes of  
3 the sub-tree.

1 21. (Previously Presented) The method of claim 19, wherein execution of the first set of  
2 rebalancing operations includes generating a list of pointers to nodes requiring  
3 updating in the second phase of rebalancing.

1 22. (Previously Presented) A method of rebalancing a tree data structure, the method  
2 comprising:  
3 allowing a sub-tree of the tree data structure to grow unbalanced until a threshold  
4 level is reached;  
5 executing a first set of rebalancing operation tasks during a first rebalancing  
6 phase, the first rebalancing phase being characterized by navigation  
7 between nodes of the sub-tree; and  
8 executing a second set of rebalancing operation tasks during a second rebalancing  
9 phase, the second rebalancing phase including navigation to two or more  
10 nodes of the sub-tree, the navigation being independent of pointers  
11 between nodes of the sub-tree.

1 23. (Previously Presented) The method of claim 22, wherein the first set of rebalancing  
2 operation tasks are performed on unlocked nodes of the sub-tree.

1 24. (Previously Presented) The method of claim 22, wherein the first set of rebalancing  
2 operation tasks includes a plurality of operation tasks configured for parallel  
3 execution.

1 25. (Previously Presented) A method of maintaining a tree data structure, the method  
2 comprising:  
3 allowing the tree data structure to grow unbalanced;  
4 performing a first set of rebalancing operation tasks during a first rebalancing  
5 phase on a plurality of nodes in the tree data structure, the first set of  
6 rebalancing operation tasks being configured for execution while the  
7 plurality of nodes are unlocked and for insertion and deletion of nodes;  
8 and  
9 performing a second set of rebalancing operation tasks on the plurality of nodes in  
10 a second rebalancing phase, the second set of rebalancing operation tasks  
11 being different than the first set of rebalancing operation tasks and being  
12 configured for further operations on the plurality of nodes, the second  
13 rebalancing phase occurring after completion of the first rebalancing  
14 phase.

1 26. (Previously Presented) The method of claim 25, wherein the first set of rebalancing  
2 operation tasks are performed in parallel.

1 27. (Previously Presented) A method comprising:  
2 storing a tree data structure;  
3 tracking the execution of operations upon the tree data structure; and  
4 rebalancing the tree data structure when an unbalanced sub-tree of the tree data  
5 structure reaches a threshold level greater than one, the rebalancing  
6 including a first rebalancing phase in which rebalancing operations are

7           executed in parallel and nodes of the unbalanced sub-tree are unlocked,  
8           and a second rebalancing phase in which different rebalancing operations  
9           are executed.

10

11   28. (Previously Presented)   The method of claim 10, wherein the first and second set of  
12   element state transitions each include changing pointers to nodes of the tree data  
13   structure.